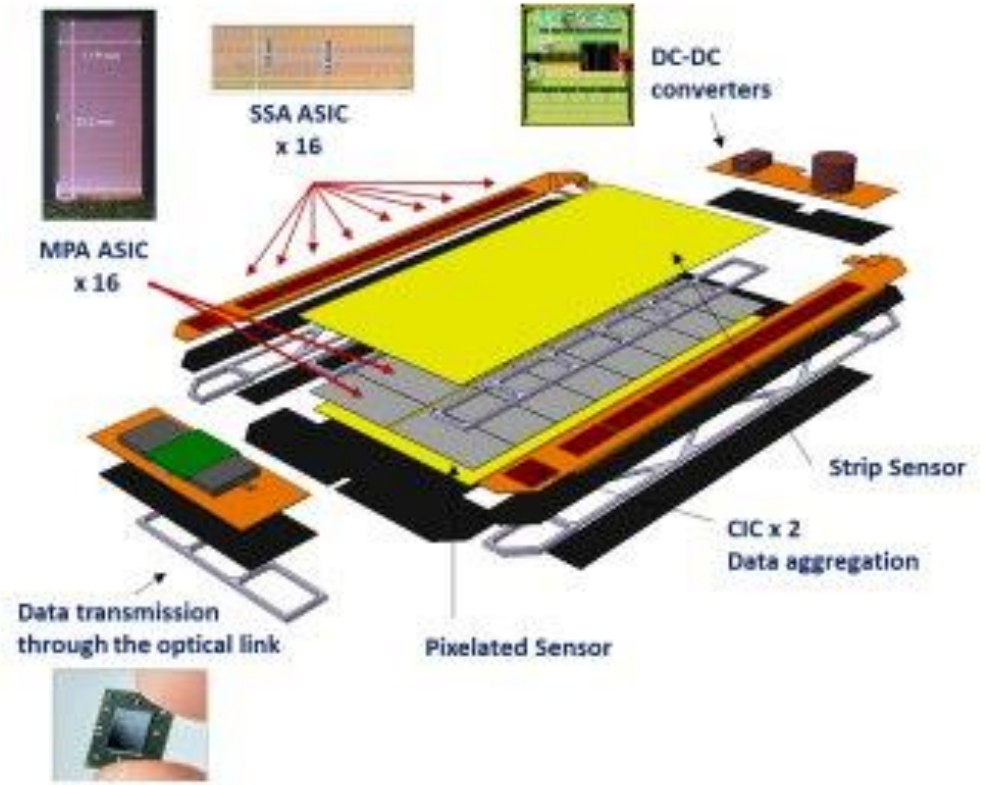
Embedded Software Application for System on Chip (SoC)



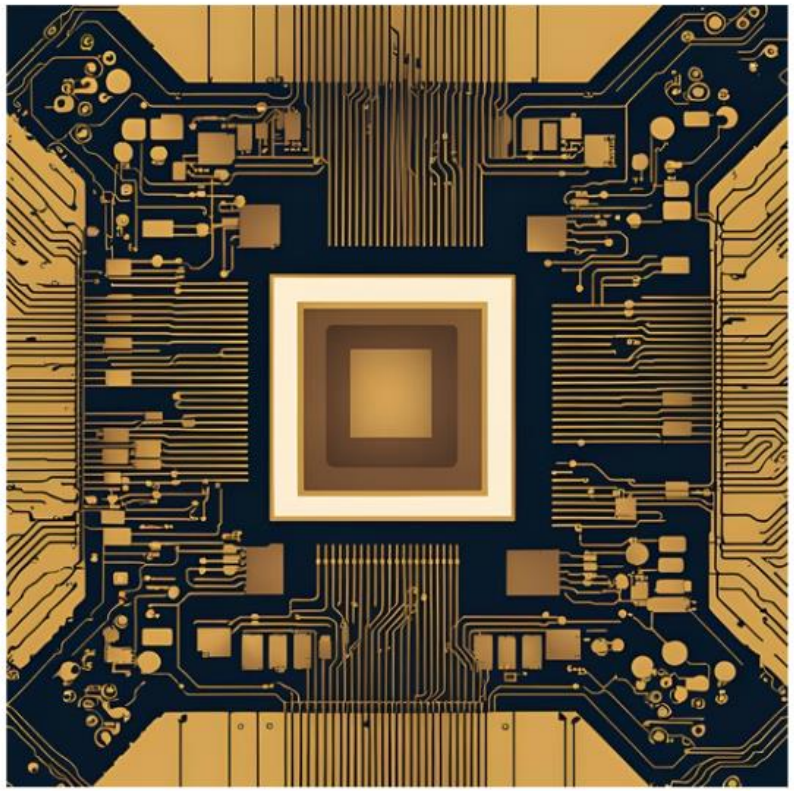
Based on RISC-V architecture for LHCb Velo detector

Author: ViktoriiaTarasenko (Kyiv Polytechnic Institute)

Mentors: Alessandro Caratelli (CERN)
Marco Andorno (CERN)

PROJECT OBJECTIVE

❖In frame of this project, we developed a technological concept how to interact with hardware abstraction layer (HAL) of the TRIGLAV microcontroller using C++ coding.

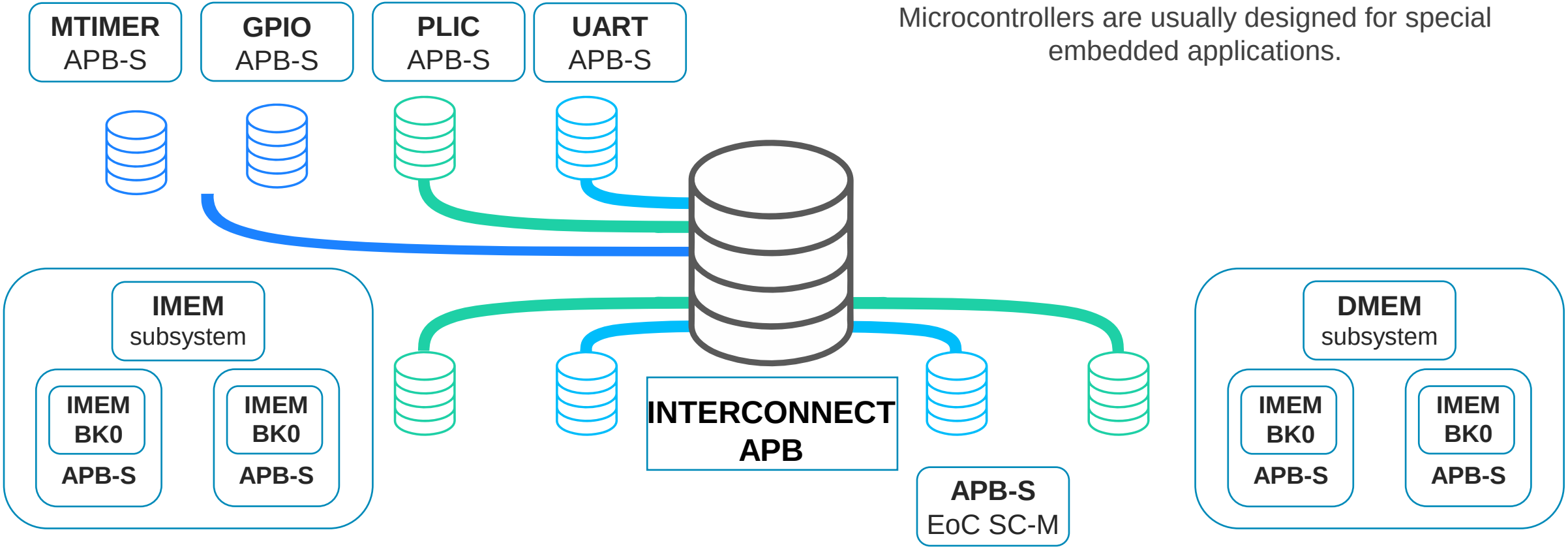


Complexity of ASIC design on a single chip

Our current focus is on integrating System-on-Chip (SoC) techniques to efficiently consolidate multiple functions onto a single chip while enabling programmability on-detector ASICs.

TRIGLAV

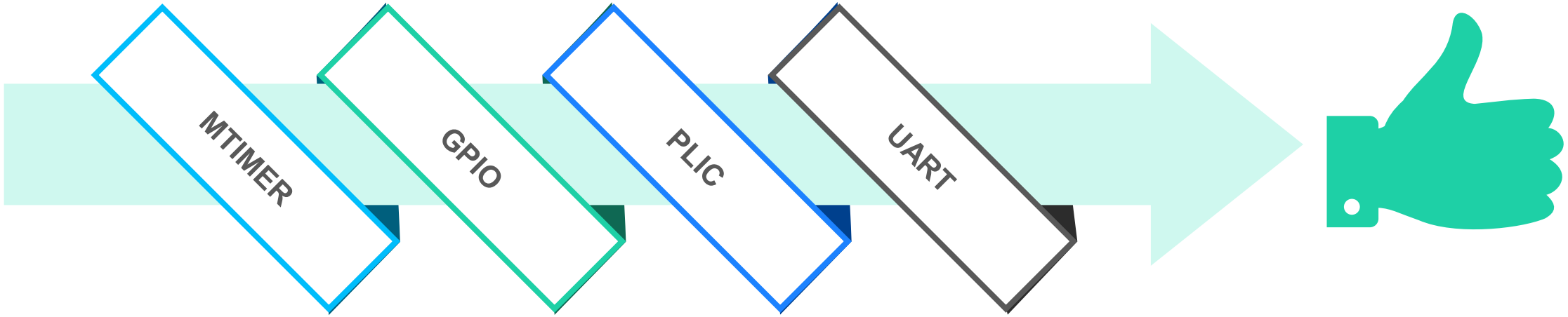
A **System on Chip (SoC)** may include a microcontroller as one of its components but usually integrates it with advanced peripherals. Microcontrollers are usually designed for special embedded applications.



Peripherals

TIMER
Provides a configurable number of 64-bit counters where each counter increments by a step value

PLIC
The PLIC module is designed to manage various interrupt sources from the peripherals.



GPIO
Provide the interface with external modules, for example sensors.

UART
It uses a serial format, meaning data is sent one bit at a time, and has a programmable speed.

SoC Automation

Merge the C++ code with the
Hardware Abstraction Layer (HAL)

Minimize the user effort in
hardware/software code design

Fast prototype reducing chances of
hardware/software bugs

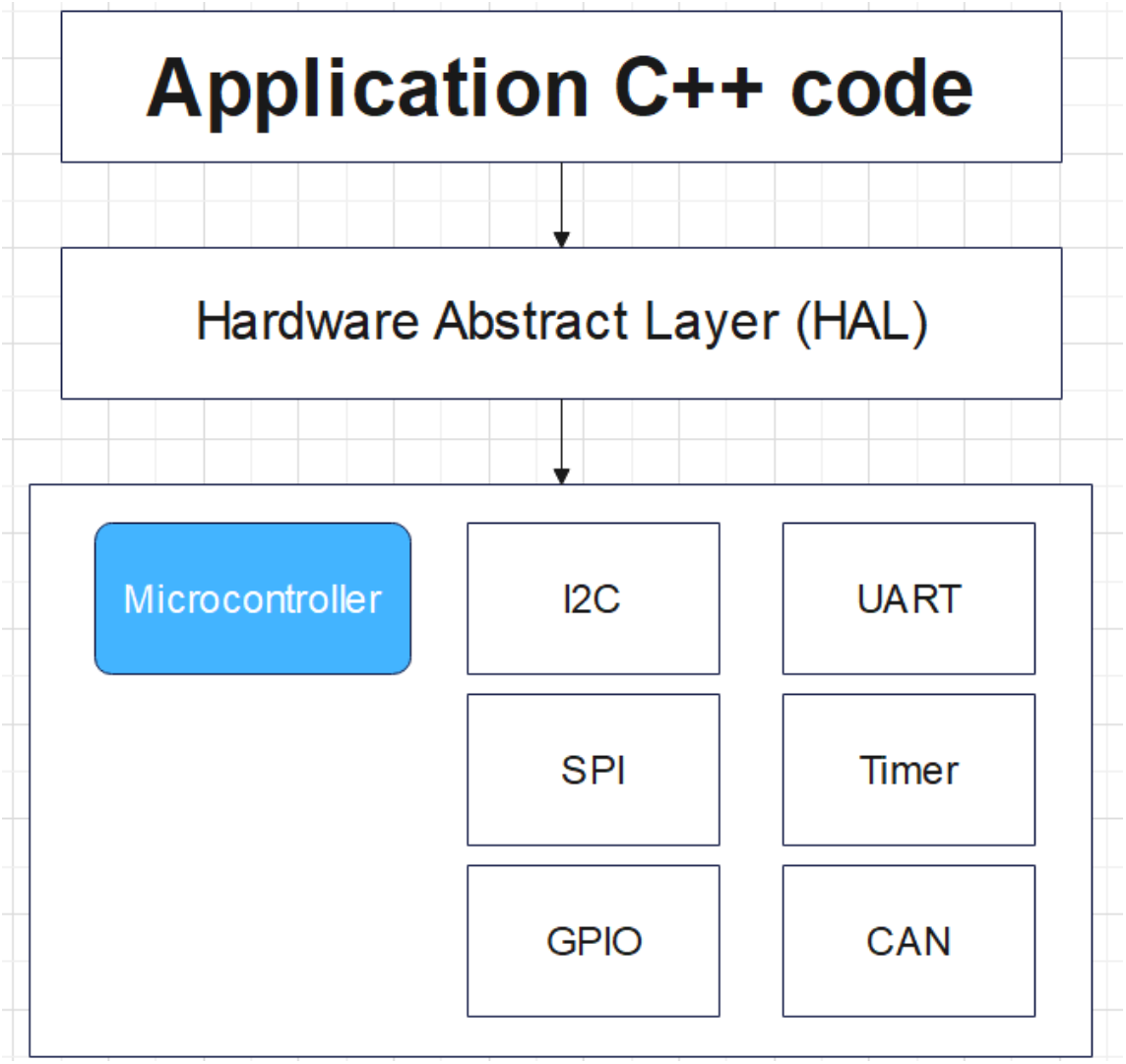
Execute applications (C/C++) on the
simulated RISC-V based SoC

Hardware Abstract Layer

A HAL is a software library that provides a common interface between the application code and the hardware-specific drivers. It simplifies the programming process by hiding the details and variations of the hardware from the application level.

After exploration of documentation such as HAL Basic Functions for GPIO:

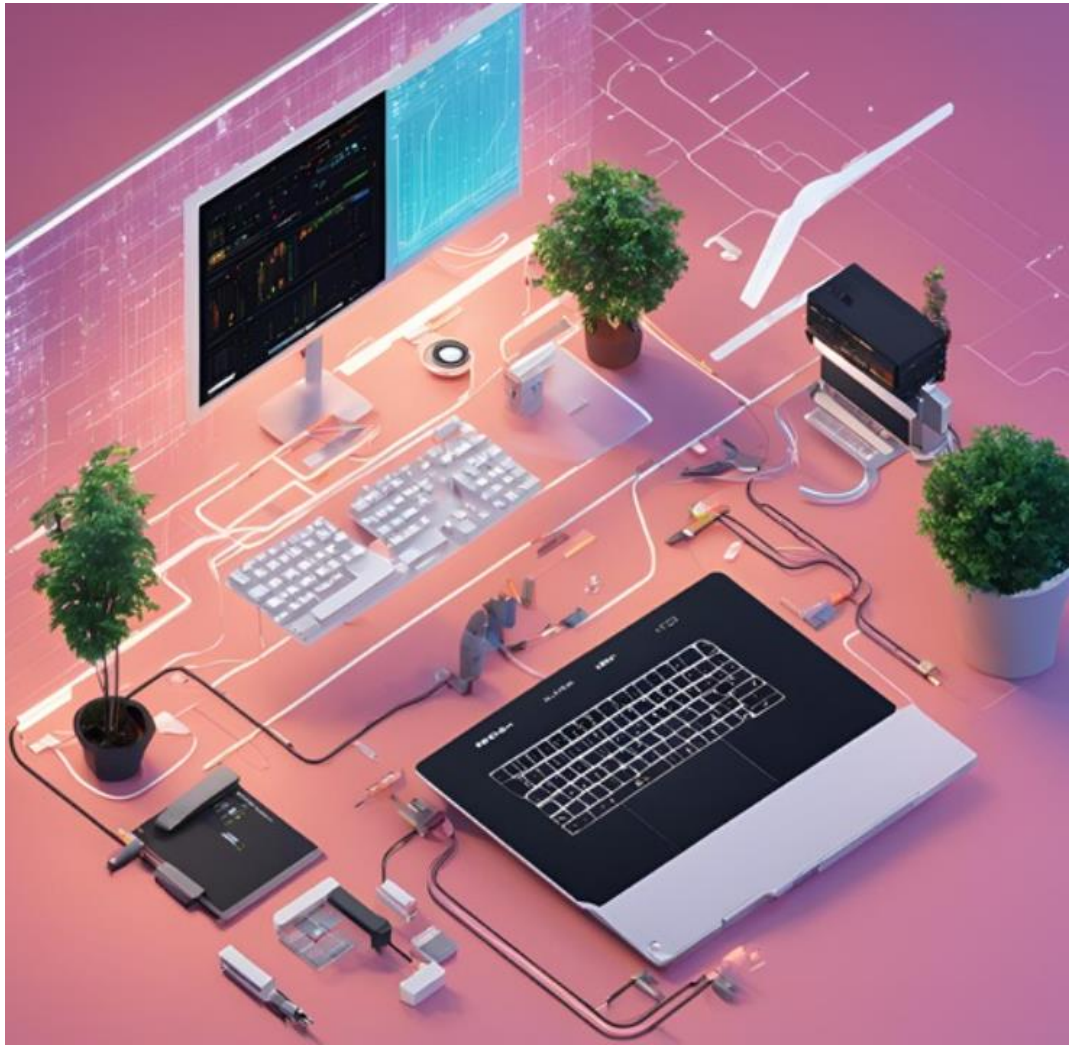
HAL_GPIO_ReadPin()
HAL_GPIO_WritePin()
HAL_GPIO_TogglePin()
HAL_GPIO_LockPin()
HAL_GPIO_EXTI_IRQHandler()



Signal for UART

For our task with the UART peripheral, we assess the UART API from HAL, which goes to the UART peripheral inside the microcontroller, where we can perform tasks like TX, RX, etc..

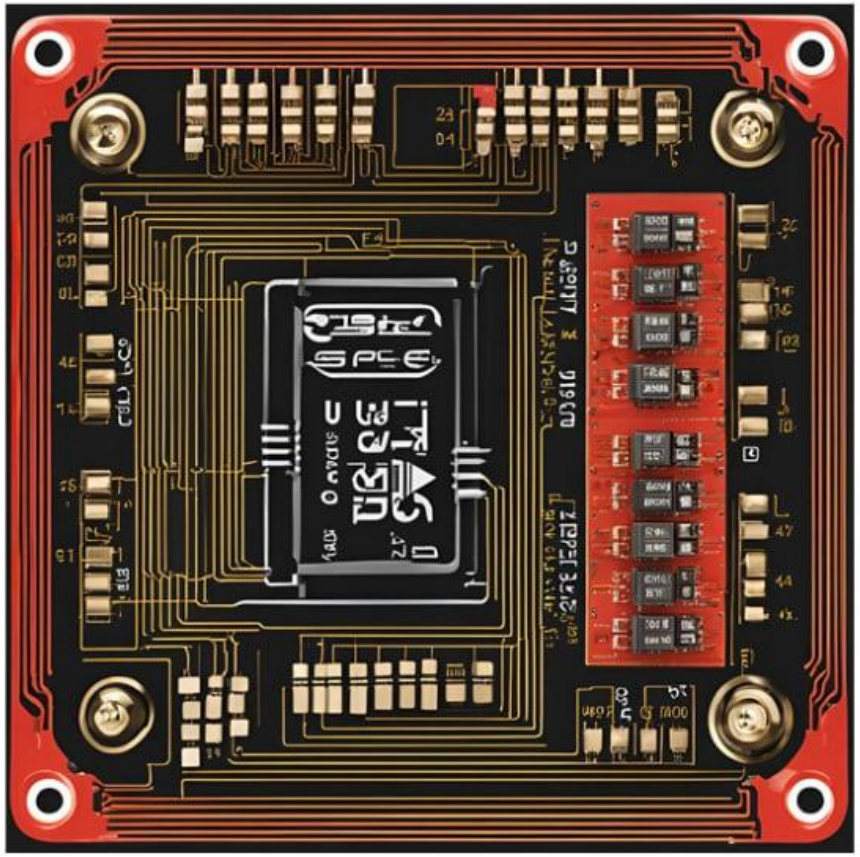
In our C++ code, we used to swap the TX pin from 1 to 0 for testing a relay. A special function can handle this using a counter to calculate it, based on time period specified in milliseconds.



GPIO Configuration

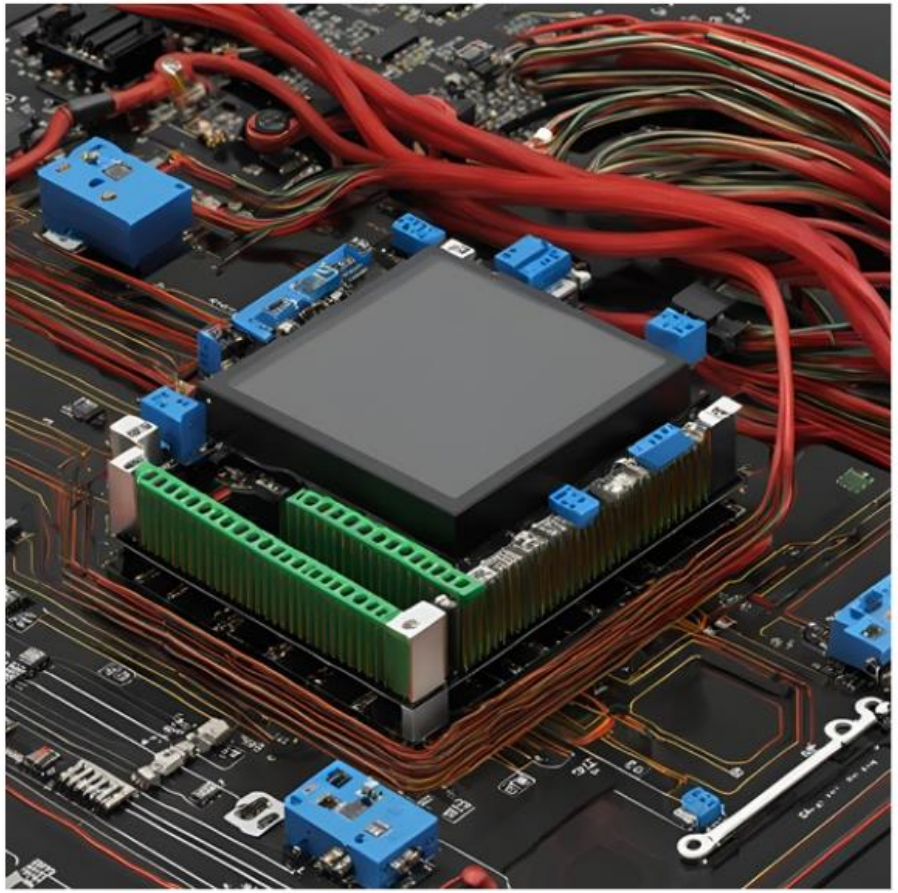
The configuration register allows selection of the GPIO in Input (bit=0) or Output (bit=1) mode by setting appropriate bits in the register.

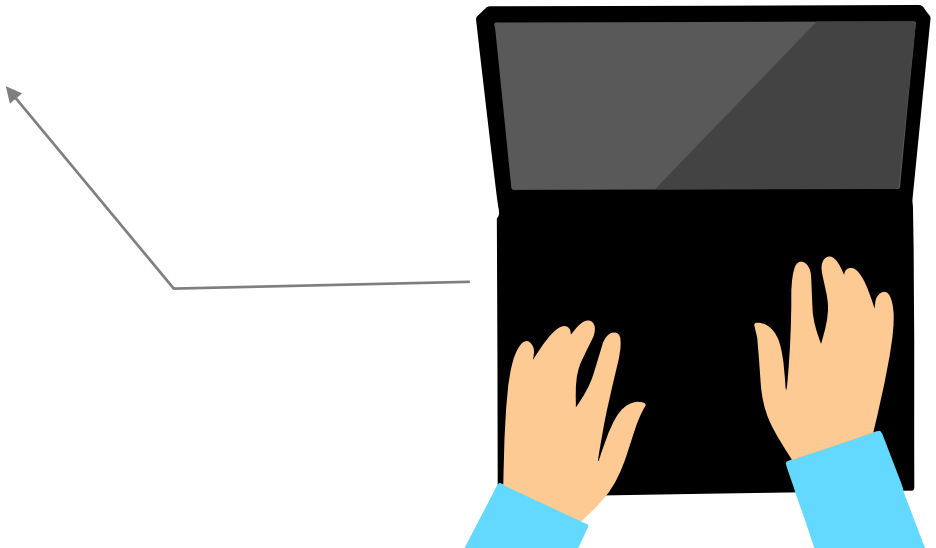
GPIO pins can be:
* Configured to be input or output;
* Enabled/disabled;
* Input values are readable and can be optionally used as interrupts;
* Output values are readable/writable;



Production

Hence, I've created a special function to measure GPIO relays of read/write cycles for ongoing evaluation and further hardware optimization. Hopefully, it can accelerate design process of the SoC automation and slightly improve the project pipeline.





Thank you!